International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Study of LMS Algorithm Using Adaptive Filtering Technique

L. Yamini Swathi[1], K. S. S. Soujanya Kumari[1], N. P. Lavanya Kumari[2], M. Purnachandra Rao[3]

[1]Teaching Associate, Department Of Systems Design, Andhra University
[2]Assistant Professor(C), Department Of Systems Design, Andhra University
[2]Research Guide, Department of Systems Design, Andhra University

Abstract: ***Many filter design techniques in Digital Signal Processing applications were based on second order statistics which include channel equalization, echo cancellation and system modeling. In these applications filters with adjustable coefficients, called Adaptive Filters were employed. Such Filters incorporate algorithms that allow the filter coefficients to adapt to signal statistics. Adaptive filtering techniques are used in a wide range of applications including echo cancellation, linear prediction, adaptive equalization, adaptive noise cancellation and adaptive beam forming. The design of adaptive filter includes i) Determination of Cost functions like Minimum Square Error (MSE) criterion and exponentially weighted Least Square Error criterion. ii) The performance of adaptive filtering algorithm which depends on the factors like Rate of convergence, misadjustment, tracking capability, computational requirement, and numerical robustness. iii) Structure determination which is inter related with the algorithm. Four common structures namely direct, parallel, cascade and lattice form structures were used. Here in this present paper the basic Least Mean Square Algorithm which is based on gradient optimization for determining the coefficients was observed. We considered the basic Widrow's Least Mean Square Algorithm in which we study optimization criterion, Adaption procedure and Performance Analysis. The two important Performance measures in LMS algorithms are Rate of Convergence and Misadjustment.***

Keywords: Digital Signal Processing, Adaptive Filters, Cost functions, Least Mean Square Algorithm

1. Introduction

Discrete-time (or digital) filters are ubiquitous in today's signal processing applications. Filters are used to achieve desired spectral characteristics of a signal, to reject unwanted signals, like noise or interferers, to reduce the bit rate in signal transmission, etc. The notion of making filters adaptive, i.e., to alter parameters (coefficients) of a filter according to some algorithm, tackles the problems that we might not in advance know, e.g., the characteristics of the signal, or of the unwanted signal, or of a systems influence on the signal that we like to compensate. Adaptive filters can adjust to unknown environment, and even track signal or system characteristics varying over time [5].

Adaptive filtering techniques are used in a wide range of applications, including echo cancellation, adaptive equalization, adaptive noise cancellation, and adaptive beam forming [1]. Under this condition, a significant improvement in performance can be achieved by using adaptive rather than fixed filters. An adaptive filter is a self-designing filter that uses a recursive algorithm (known as adaptation algorithm or adaptive filtering algorithm) to "design itself." The algorithm starts from an initial guess, chosen based on the a priori knowledge available to the system, then refines the guess in successive iterations, and converges, eventually, to the optimal Wiener solution in some statistical sense [3]. An Adaptive Filter is a time-variant filter whose coefficients are adjusted in a way to optimize a cost function or to satisfy some predetermined optimization criterion.

2. Methods

Characteristics of adaptive filters are- they can automatically adapt (self-optimize) in the face of changing environments and changing system requirements They can be trained to perform specific filtering and decision-making tasks according to some updating equations (training rules). Adaptive filters are used because they can automatically operate in changing environments (e.g. signal detection in wireless channel), non-stationary signal/noise conditions (e.g. LPC of a speech signal) and time-varying parameter estimation (e.g. position tracking of a moving source) [2].

Block diagram of a typical Adaptive filter is shown below

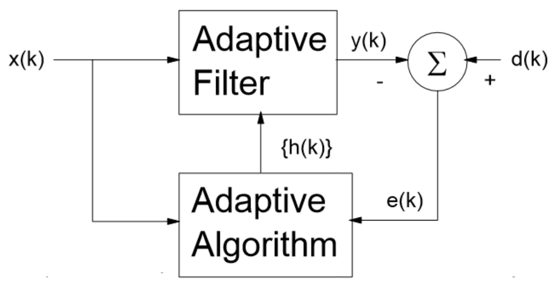

Figure 1(a)

From the above figure 1(a), x (k) is input signal; y (k) is filtered output; d (k) is desired response and h (k) is impulse response of adaptive filter. Here the cost function is defined as **E {e^2 (k)}** or $\sum_{k=0}^{N-1}$ e^2 (k)

2.1 Design Considerations of the Adaptive Filter Include

2.1.1 Cost Function:

Choice of cost functions depends on the approach used and the application of interest [6]. Some commonly used cost functions are

Mean square error (MSE) criterion: minimizes **E{e²(k)}** where E denotes expectation operation, e(k)=d(k)-y(k) is the estimation error, d(k) is the desired response and y(k) is the actual filter output

Exponentially weighted least squares criterion: Minimizes $\sum_{k=0}^{N-1} \lambda^{N-k-1} \mathbf{e2(k)}$ where N is the total number of samples and λ denotes the exponentially weighting factor whose value is positive close to 1.

2.1.2 Algorithm

Depends on the cost function used and

i) **Rate of convergence:** This corresponds to the time required for the algorithm to converge to the optimum least squares/Wiener solution.

ii) **Misadjustment:** Excess mean square error (MSE) over the minimum MSE produced by the Wiener filter, mathematically it is defined as

$$M = \frac{\lim_{k\to\infty} E\{e^2(k)\} - \varepsilon_{\min}}{\varepsilon_{\min}}$$

iii) **Tracking capability:** This refers to the ability of the algorithm to track statistical variations in a non-stationary environment.

iv) **Computational requirement:** It is the number of operations, memory size, investment required to program the algorithm on a computer.

v) **Numerical robustness:** This refers to the ability of the algorithm to operate satisfactorily with ill-conditioned data, e.g. very noisy environment, change in signal and/or noise models

2.1.3 Structure

Structure and algorithm are inter-related. Choice of structures is based on quantization errors, ease of implementation, computational complexity, etc. Four commonly used structures are **direct form, cascade form, parallel form,** and **lattice structure [4]**. Advantages of lattice structures include simple test for filter stability, modular structure and low sensitivity to quantization effects.

2.2 Widrow's Least Mean Square (LMS) Algorithm

The study of LMS algorithm is based upon Optimization criterion, Adaption Procedure and Performance Analysis.

2.2.1. Optimization Criterion

This is used to minimize the **mean square error E {e² (n)}**

2.2.2. Adaptation Procedure

It is an approximation of the steepest descent method where the expectation operator is ignored, i.e.

$\frac{\partial E\{e^2(n)\}}{\partial \underline{W}(n)}$ is replaced by $\frac{\partial e^2(n)}{\partial \underline{W}(n)}$

The LMS algorithm is therefore:

$$\begin{aligned}
\underline{W}(n+1) &= \underline{W}(n) - \mu \frac{\partial e^2(n)}{\partial \underline{W}(n)} \\
&= \underline{W}(n) - \mu \frac{\partial e^2(n)}{\partial e(n)} \cdot \frac{\partial e(n)}{\partial \underline{W}(n)} \\
&= \underline{W}(n) - 2\mu e(n) \cdot \frac{\partial\left[d(n) - \underline{W}^T(n) \cdot \underline{X}(n)\right]}{\partial \underline{W}(n)}, \qquad \frac{\partial \underline{A}^T \cdot \underline{B}}{\partial \underline{A}} = \underline{B} \\
&= \underline{W}(n) + 2\mu e(n)\underline{X}(n)
\end{aligned}$$

or

$$w_i(n+1) = w_i(n) + 2\mu e(n)x(n-i), \qquad i = 0,1,\cdots,L-1$$

2.2.3. Performance Analysis

Two important performance measures in LMS algorithms are rate of convergence & misadjustment (relates to steady state filter weight variance).

1. Convergence Analysis:

For ease of analysis, it is assumed that $\underline{W}$ (n) is independent of $\underline{X}$ (n).

Taking expectation on both sides of the LMS algorithm, we have

$$\begin{aligned}
E\{\underline{W}(n+1)\} &= E\{\underline{W}(n)\} + 2\mu E\{e(n)\underline{X}(n)\} \\
&= E\{\underline{W}(n)\} + 2\mu E\left\{d(n)\underline{X}(n) - \underline{X}(n)\cdot(\underline{X}^T(n)\underline{W}(n))\right\} \\
&= E\{\underline{W}(n)\} + 2\mu \underline{R}_{dx} - 2\mu \underline{R}_{xx} E\{\underline{W}(n)\} \\
&= (\underline{I} - 2\mu \underline{R}_{xx})E\{\underline{W}(n)\} + 2\mu \underline{R}_{xx} \underline{W}_{MMSE}
\end{aligned}$$

Which is very similar to the adaptive equation in the steepest descent method.

Following the previous derivation, W (n) will converge to the Wiener filter weights in the mean sense if

$$\begin{bmatrix}
\lim_{n\to\infty}(1-2\mu\lambda_1)^n & 0 & \cdots & 0 \\
0 & \lim_{n\to\infty}(1-2\mu\lambda_2)^n & & \vdots \\
\vdots & & \ddots & 0 \\
0 & \cdots & 0 & \lim_{n\to\infty}(1-2\mu\lambda_L)^n
\end{bmatrix} = \underline{0}$$

$$\Rightarrow |1-2\mu\lambda_i| < 1, \qquad i = 1,2,\cdots,L$$

$$\Rightarrow 0 < \mu < \frac{1}{\lambda_{\max}}$$

Define geometric ratio of the pth term as

$$r_p = 1 - 2\mu\lambda_p, \qquad p = 1,2,\cdots,L$$

It is observed that each term in the main diagonal forms a geometric series

$$\{1, r_p^1, r_p^2, \cdots, r_p^{n-1}, r_p^n, r_p^{n+1}, \cdots\}.$$

Exponential function can be fitted to approximate each geometric series:

$$r_p \approx \exp\left(-\frac{1}{\tau_p}\right) \Rightarrow \{r_p^n\} \approx \left\{\exp\left(-\frac{n}{\tau_p}\right)\right\}$$

where τ_p is called the p th time constant.

For slow adaptation, i.e. $2\mu\lambda_p << 1$, τ_p is approximated as

$$-\frac{1}{\tau_p} = \ln(1-2\mu\lambda_p) = -2\mu\lambda_p - \frac{(-2\mu\lambda_p)^2}{2} + \frac{(-2\mu\lambda_p)^3}{3} - \cdots \approx -2\mu\lambda_p$$

$$\Rightarrow \tau_p \approx \frac{1}{2\mu\lambda_p}$$

Notice that the smaller the time constant the faster the convergence rate. Moreover, the overall convergence is limited by the slowest mode of convergence which in turns stems from the smallest eigenvalue of R_{xx}, λ_{min}

That is,

$$\tau_{max} \approx \frac{1}{2\mu\lambda_{min}}$$

In general, the rate of convergence depends on two factors:

i) Step size μ: the larger the μ, the faster the convergence rate

ii) Eigenvalue spread of R_{xx}, $\chi(\underline{R}_{xx})$ **:** the smaller $\chi\underline{R}_{xx}$, the faster the convergence rate. $\chi\underline{R}_{xx}$ is defined as

$$\chi(\underline{R}_{xx}) = \frac{\lambda_{max}}{\lambda_{min}}$$

Notice that $1 \le \chi\underline{R}_{xx} < \infty$. It is worthy to note that although χR_{xx} cannot be changed, the rate of convergence will be increased if we transform x(n) to another sequence, say, y(n), such that $\chi\underline{R}_{xx}$ is close to 1.

3. An Illustration of Eigen Value Spread for LMS Algorithm

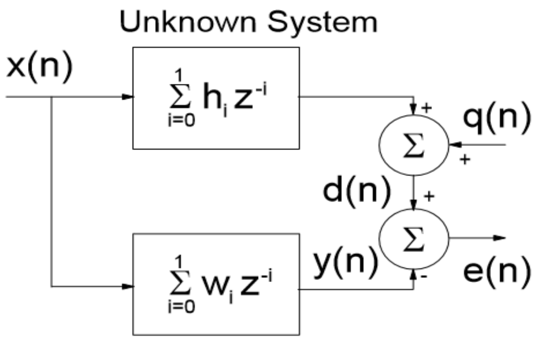

Figure 1(b)

From fig 1 (b), we have, d (n) = h_0x (n) + h_1x (n-1) + q (n)
y (n) = w_0 (n) x(n) + w_1(n) x(n-1)
e(n) = d(n) – y(n) = d(n) – w_0(n)x(n) – w_1(n)x(n-1)
w_0 (n+1) = w_0 (n) + 2u e (n) x (n)
w_1 (n+1) = w_1 (n) + 2u e (n) x (n-1)

3.1 The following is a sample program to illustrate the Eigen value spread for LMS algorithm

```
Clear all
N=1000; % number of sample is 1000
np = 0.01; % noise power is 0.01
sp = 1; % signal power is 1 which implies SNR = 20dB
h= [1 2]; % unknown impulse response
x = sqrt (sp).*randn (1, N);
d = conv(x, h);
d = d (1: N) + sqrt (np).*randn (1, N);
w0 (1) = 0; % initial filter weights are 0
w1 (1) = 0;
mu = 0.005; % step size is fixed at 0.005
y (1) = w0 (1)*x (1); % iteration at "n=0"
e (1) = d (1) - y(1); % separate because "x(0)" is not defined
w0 (2) = w0 (1) + 2*mu*e(1)*x(1); w1(2) = w1(1);
for n=2: N % the LMS algorithm
y (n) = w0 (n)*x(n) + w1(n)*x(n-1);
e (n) = d (n) - y(n);
w0 (n+1) = w0 (n) + 2*mu*e (n)*x (n);
w1 (n+1) = w1 (n) + 2*mu*e (n)*x (n-1); end
n = 1: N+1;
subplot (2, 1, 1)
plot (n, w0) % plot filter weight estimate versus time
axis ([1 1000 0 1.2])
subplot (2, 1, 2)
plot (n, w1)
axis ([1 1000 0 2.2])
figure (2)
subplot (1, 1, 1)
n = 1: N; semilogy (n, e.*e); % plot square error versus time
```

4. Results

From the following MATLAB Program we considered the number of samples equal to 1000 and signal power is 1.We get the output basing on the weight functions w_0 and w_1. They both converge at same speeds as mentioned below. Here n is the time period.

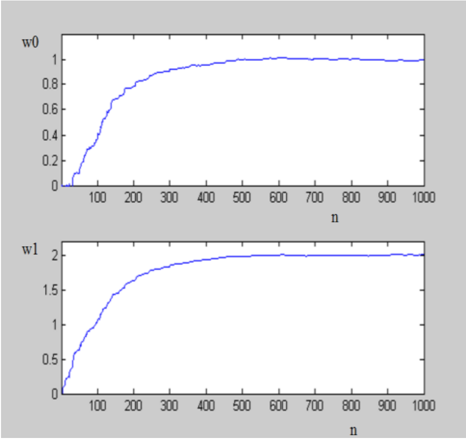

Figure 1(c) & (d)

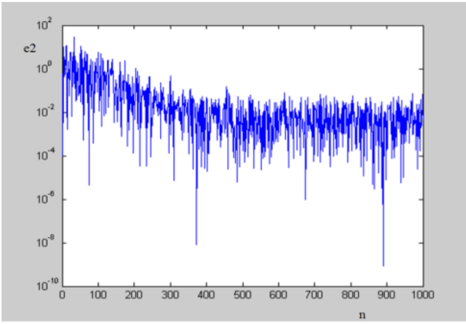

Figure 1(e)

From Figures 1(c) & 1(d) we can observe that both filter weights w_0 and w_1 converge at similar speed with respect to time n because the eigenvalues of the $\underline{R}_{xx}$ are identical. From fig 1(e) depicts the Eigen value spread which is equal to unity and converges with same speed.

We know that

$$R_{xx} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{bmatrix}$$

For white process with unity power, we have

$$R_{xx}(0) = E\{x(n).x(n)\} = 1$$
$$R_{xx}(1) = E\{x(n).x(n-1)\} = 0$$

As a result,

$$R_{xx} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \; \chi(\underline{R}_{xx}) = 1$$

5. Future Scope

Since the Eigen value spread of Newton based approach is 1, we can combine the LMS algorithm and Newton's Method to form "LMS/Newton" algorithm as follows,

$$\underline{W}(n+1) = \underline{W}(n) - \frac{\mu}{2}\underline{R}_{xx}^{-1}\frac{\partial e^2(n)}{\partial \underline{W}(n)}$$
$$= \underline{W}(n) + \mu \underline{R}_{xx}^{-1} e(n)\underline{X}(n)$$

The result will be as follows

i) The computational complexity of the LMS/Newton algorithm is smaller than the RLS algorithm but greater than the LMS algorithm.

ii)When $\underline{R}_{xx}$ is not available, it can be estimated as follows

$$\hat{R}_{xx}(l,n) = \alpha\hat{R}_{xx}(l,n-1) + x(n+l)\cdot x(n), \quad l = 0,1,\cdots,L-1$$

Where $\hat{R}_{xx}(l,n)$ represents the estimate of $R_{xx}(1)$ at time n and $0 < \alpha < 1$.

6. Conclusion

From the above algorithm we can conclude that if the Eigen values of $\underline{R}_{xx}$ are identical then both the filters converge at similar speed and the Eigen spread value $\chi\underline{R}_{xx}$ is unity. If the signal speed is varied the Eigen values changes accordingly and the Eigen spread value increases and the weight function also varies depending on the Eigen values.

Steepest descent method is simpler than the Newton method since no matrix inversion is required. The convergence rate of Newton method is much faster than that of the steepest descent method. However, both methods require exact values of $\underline{R}$xx and $\underline{R}_{dx}$ which are not commonly available in practical applications. This is the main advantage why we use LMS algorithm because it does not need statistics of signals, ie., $\underline{R}$xx and $\underline{R}_{dx}$. Moreover LMS computations have low computational complexity and are simple to implement and they allow real-time operation

References

[1] B.Widrow and M.E.Hoff, "Adaptive switching circuits," Proc. Of WESCON Conv. Rec., part 4, pp.96-140, 1960
[2] B.Widrow and S.D.Stearns, Adaptive Signal Processing, Prentice-Hall, 1985
[3] O.Macchi, Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission, Wiley, 1995
[4] P.M.Clarkson, Optimal and Adaptive Signal Processing, CRC Press, 1993
[5] S.Haykin, Adaptive Filter Theory, Prentice-Hall, 2002
[6] D.F.Marshall, W.K.Jenkins and J.J.Murphy, "The use of orthogonal transforms for improving performance of adaptive filters", IEEE Trans. Circuits & Systems, vol.36, April 1989, pp.474-483

Author Profile

Prof. M. Purnachandra Rao received his M.Sc (Tech) in Electronics from Andhra University 1980 and he was awarded his Ph.D. from Dept. of Physics, Andhra University. Later on appointed has Assistant Professor, Associate Professor and Professor in Physics Department and Head of the Dept. of Systems Design. Has 30 years of experience in the field of SODAR and its applications.

L. Yamini Swathi received M. Sc and M. Phil degrees in Electronics from Andhra University in 2004 and 2008. She is working as Teaching Associate in the Department of Systems Design, A.U. Visakhapatnam.

K. S. S. Soujanya Kumari received M. Sc and M. Phil degrees in Electronics from Andhra University in 2004 and 2008. She is working as Teaching Associate in the Department of Systems Design, A.U. Visakhapatnam.

N. P. Lavanya Kumari received M. Sc, M. Phil in Electronics and M. Tech degrees in Electronics & Communication Engineering from Andhra University in 2004, 2008 and 2012 respectively. She is working as an Asst. Professor(C) in the Department of Systems Design, A.U. Visakhapatnam.